Terminology

System - All files developed for the purposes of satisfying the client level requirements.

- **Data display module** - displays status updates and events to the user
- **Controller simulator** - Generates status updates and events then sends via RS422 serial protocol to the data display module.
- **Event Log file** - Will be generated by the data display module to contain 1 or more event strings encountered during a session.
- **Installer** - will deploy the project in the customers system and perform environment setup and initialization. Will record initial user preferences and take them into account during installation. I.e. “Do you want a shortcut on your desktop?”

Environment - The directory our system will be placed in and all of its contents.

Status information - (See class diagram for specifics) General data pertaining to the weapon which can be measured at any point in time during session.

Status Update - serialized version of status information to be sent to the data display module from the controller simulator.

Event String - A string of the format “<time> <event message> <param 1> <param 2> <param 3>” generated by the controller simulator to simulate the occurrence of a weapon related event. The parameters can be NULL, but the event message must be specific text outlining what the event is. Ex. “[00:12:41] Weapon overheat 237 200” where 237 represents measured barrel temp and 200 represents max recommended barrel temp in degrees celsius.

Event update - The serialized version of the event string to be sent to the data display module from the controller simulator.

Session - The time measured from the moment the controller sim is connected to the data display module to the moment the controller sim is disconnected from the data display module.

Traceability Matrix

The purpose of this table is to define which system requirements satisfy each client requirement and which software requirements/functions will be used by each system requirement.

Client requirement	System requirements	Software requirements
CR01 The data display module shall be a desktop application.	**R01** The data display module shall be an .exe file.	NA
	R02 The data display module shall run a GUI.	SR18 SR19 SR20 SR21 SR22 SR23
CR02 The data display module shall read input data via RS422 serial protocol from the controller simulator.	**R03** The **data display module** shall be capable of **deserializing** messages received via an RS422 serial port.	SR02
	R04 The **controller simulator** shall be capable of **deserializing** messages received via an RS422 serial port.	SR02
	R05 The **data display module** shall be capable of **serializing** messages to send through an RS422 serial port.	SR01
	R06 The **controller simulator** shall be capable of **serializing** messages to send through an RS422 serial port.	SR01

CR03 The data display module shall have the ability to write event data into a log file	**R07** The data display module shall be capable of generating a log file including all events up to the point in which the user requests the log file (this is the case for **generating a log file during a session**)	SR08 SR10 SR11 SR23
	R08 The data display module shall be capable of automatically generating a log file including all events that occurred during a complete session. (this is the case for **automatically generating a log file after a session has ended**)	SR08 SR10 SR11
	R09 Up to 5 auto-saved event log files shall be stored in the log file folder until overwrites occur on the oldest auto-saved file.	SR11
CR04 The data display module shall display all weapon status information directly to the application's window for the duration of a session.	**R10** The controller simulator shall send status updates through serial port every .25 seconds.	SR04 SR05 SR09
CR05 The controller simulator shall send event updates to the data display module.	**R11** The controller simulator shall send event updates through the serial port at most 0.1 second after they are generated.	SR04 SR05
CR06	**R12**	NA

The data display module shall not require admin rights to install, set up, or use.	The data display module shall not require admin rights to install setup or use	
CR07 The data display module shall include filtering options to filter events and errors	**R13** The data display module shall have the capability to display **only errors** to the Events tab of the GUI	SR13
	R14 The data display module shall have the capability to display **only cleared errors** to the events tab of the GUI.	SR13
	R15 The data display module shall have the capability to display **only active errors** to the events tab of the GUI.	SR13
	R16 The data display module shall have the capability to display only **non-error events** to the events tab of the GUI.	SR13
CR08 The system and its environment shall be installed via an installer file.	**R17** The system and its environment shall be installed via an installer file	NA
CR09 The system shall be portable on Windows 10 or 11	**R18** The system shall be portable on Windows 10 or 11	NA

CR10G The system should be portable on Debian linux distributions	**R19G** The system should be portable on Debian linux distributions	NA

Client level requirements

- **CR01** - The data display module shall be a desktop application.
- **CR02** - The data display module shall read input data via RS422 serial protocol from the controller simulator through a user specified port.
- **CR03** - The data display module shall have the ability to write event data into a log file
- **CR04** - The data display module shall display all status information directly to the application's window for the duration of a session.
- **CR05** - The controller simulator shall send event updates to the data display module.
- **CR06** - The data display module shall not require admin rights to install, set up, or use.
- **CR07** - The data display module shall include filtering options to filter events and errors
- **CR08** - The system and its environment shall be installed via an installer file.
- **CR09** - The system shall be portable on Windows 10 or 11
- **CR10G** - The system should be portable on Debian linux distributions

System Level Requirements

- **R01** - The data display module shall be an .exe file.
- **R02** - The data display module shall run a GUI.
- **R03** - The data display module shall be capable of deserializing messages received via an RS422 serial port.
- **R04** - The controller simulator shall be capable of deserializing messages received via an RS422 serial port.
- **R05** - The data display module shall be capable of serializing messages to send through an RS422 serial port.

- **R06** - The controller simulator shall be capable of serializing messages to send through an RS422 serial port.
- **R07 -** The data display module shall be capable of generating a log file including all events up to the point in which the user requests the log file (this is the case for generating a log file during a session)
- **R08** - The data display module shall be capable of automatically generating a log file including all events that occurred during a complete session. (this is the case for automatically generating a log file after a session has ended)
- **R09** - Up to 5 auto-saved event log files shall be stored in the log file folder until overwrites occur on the oldest auto-saved file.
- **R10** - The controller simulator shall send status updates through serial port every .25 seconds.
- **R11** - The controller simulator shall send event updates through the serial port at most 0.1 second after they are generated.
- **R12** - The data display module shall not require admin rights to install setup or use
- **R13 -** The data display module shall have the capability to display **only errors** to the Events tab of the GUI
- **R14** - The data display module shall have the capability to display **only cleared errors** to the events tab of the GUI.
- **R15** - The data display module shall have the capability to display **only active errors** to the events tab of the GUI.
- **R16** - The data display module shall have the capability to display only **non-error events** to the events tab of the GUI.
- **R17 -** The system and its environment shall be installed via an installer file
- **R18** - The system shall be portable on Windows 10 or 11
- **R19G** - The system should be portable on Debian linux distributions

Software Level Requirements

- **SR01 -** The functions serializeStatus() and serializeEvent() shall be capable of generating bit string versions of given status data and event data respectably.
- **SR02 -** The functions deserializeStatus() and deserializeEvent() shall be capable of generating the original status data and event data, given a serialized bit string.
- **SR03 -** The connect() function shall be capable of performing handshake protocols with another device running connect().
- **SR04 -** The function sendMessage() shall be able to transmit a serialized string through the serial port.
- **SR05 -** The function readMessage() shall be able to detect incoming serialized messages from the serial port and update the status class with new data.
- **SR06** - The function disableComs() shall be able to pause serial communication.
- **SR07 -** The function enableComs() shall be able to resume serial communication.
- **SR08 -** The function initializeEventLL() shall be capable of creating a head node for a linked list given an event string.
- **SR09** - The function initializeElectricalLL() shall be capable of creating a head node for a linked list given a set of electrical data. Then adding an unknown number of new nodes for subsequent electrical components.
- **SR10** - The function addEvent() shall be capable of adding a new event to the event linked list.
- **SR11 -** The function outputEventLog() shall be capable of writing a log file, given a linked list of event data.
- **SR12G -** The function outputElectricalLog() shall be capable of writing a log file, given a linked list of electrical component data.
- **SR13 -** The function filterEvents() shall be capable of filtering event data displayed in the GUI according to user preference.

- **SR14 -** The function randomizeEvents() shall be capable of generating random event strings out of a given set of events at random time intervals within a given range.
- **SR15 -** The function randomizeStatus() shall be capable of generating random status updates every 0.1 seconds.
- **SR16 -** The function customEvent() shall be capable of taking in user input data and generating an event string using this data.
- **SR17 -** The function customStatus() shall be capable of taking in user input data and updating the status data structure using this data.
- **SR18 -** The function getInput() shall be capable of listening for user input data and deciding what to do with it (for controller simulator).
- **SR19 -** The Function clickEvents() shall be capable of redirecting the user to the events page.
- **SR20 -** The Function clickStatus() shall be capable of redirecting the user to the status page.
- **SR21 -** The Function clickElectrical() shall be capable of redirecting the user to the electrical page.
- **SR22 -** The Function clickSettings() shall be capable of redirecting the user to the settings/help page.
- **SR23 -** The Function dropboxFilter() shall be a dropdown box allowing the user to select the correct filter.
- **SR24 -** The Function clickDownload() shall allow the user to download all the program's content into a log file.